Autonomous Synthesis of Goal-oriented Behaviors for Planetary Robotic Sampling

Sukhan Lee[1,2] *Paul S. Schenker*[1]

Jet Propulsion laboratory[1]
California Institute of Technology
Pasadena, CA 91109

Dept. of ItJ+-Systems and Computer Science[2]
University of Southern California
Los Angeles, CA 90089-0781

Abstract

Au intelligent robotic architecture that autonomously synthesizes goal-oriented behaviors, while connecting sensing and action in real-time, is presented with applications to loosely defined planetary sampling missions. By the goal-oriented behaviors, we mean sequences of actions generated from automatic task monitoring and replanning toward set goals in the presence of uncertainties as well as errors ant] faults.

This architecture is composed of perception and action nets interconnected in closed loops. The perception net, represented as a hierarchy of features that can bc extracted from physical as well as logical sensors, manages uncertainties with sensor fusion, sensor planning, and consistency maintenance. The action net, represented as a hierarchy of state transition in which all the possible system behaviors arc embedded, generates robust and fault-tolerant system behaviors with on-line adaptive task monitoring and replanning. The proposed intelligent robotic architecture is significant for autonomous planetary robotic sampling - and related robotic tasks in unstructured environments - that require robust and fault tolerant behaviors due to expected uncertainties as well as errors in sensing, actuation, and environmental constraints. We use a typical Mars planetary sampling scenario to evaluate the proposed architecture: autonomous soil science where a robot arm trenches soil to examine and deposit soil samples to lander base'cl science instrumentation.

keywords: Perception Net, Action Net, Intelligent Robotic Architecture, Goal-Oriented Behaviors, Planetary Sampling

1 Introduction

Robotic systems aim at achieving intelligence in behavior and dexterity in motion through a real-time connection between sensing and action. To achieve such intelligence and dexterity often requires an integration of distributed sensors aud actuators, such that a rich source of sensory and movement patterns that can be clustered into higher levels of concepts and actions can be provided. The key to successful integration may be a system architecture that supports com putational requirement s unique to robotics, including uncertainty management and adaptive error recovery through the interaction among such processes as feature transformation and abstraction, data and concept fusion[1, 2, 3, 4, 5, 6], consistency maintenance among dat a and knowledge[7], as well as monitoring and replanning.

In spite of the fact that a decade of research and development in robotics has produced numerous theoretical and experimental results, robots arc yet to acquire the level of intelligence and dexterity required for autonomous task execution in unstructured environments. Conventional approaches for building robotic systems without underlying computational principles of integrating sensing, knowledge, ant] action in real-t,irilc seem to suffer from the limitation in task complexity it can handle. Should robot intelligence be measured in terms of a ~)ower-to-weight ratio, where the power is defined by the product of the complexity and execution speed of tasks and the weight is defined by the product of volume and cost associated with the required hardware and software, an order of magnitude of improvement in the power-to-weig]lt ratio seems necessary for the new generation of robotics. Robot intelligence may be manifested by its extended autonomy. However, the extension should not simply be the result of aggregating additional functional units, which may cause the reduction of power or

power-to-wcig'llt ratio by increasing space and time complexity. It is necessary to develop a system architecture that supports extended autonomy without decrease in the power or power-to-weight ratio. An architecture which embeds system knowledge as well as a general problem-solving paradigm in itself may be desirable.

Planetary robotic science sampling represents in-situ analysis and collection of surface and subsurface soil ancl rock samples by robots, for the analysis of their chemical and mineral compositions with science instruments as well as for the measurement of their geological, mechanical, and thermal properties with appropriate sensors. Robots engaged in planetary science sampling (e.g., on Mars) should be capable of autonomously handling and operating science instruments and sensors, trenching and scooping soil, as well as manipulating, drilling, cutting, and collecting rocks. Planetary science sampling robots arc required to carry out loosely defined robotic missions and tasks to deal with uncertainties from noisy and uncalibrated sensors, unexpected events from unknown environments, system faults from possible hardware and software failures, and system constraints from the limited resources in power, weight, computation, sensing, and actuation.

Therefore, planetary science sampling robots should possess extended autonomy with the capabilities of uncertainty management, adaptation to ncw situations, and fault tolerance. To provide the robots with extended autonomy requires the integration of a high level of discrete event planning and a low level of continuous time control in a hierarchy of lnulti-resolution time scales. However, such integration should be done under the limitation of computational power and the requirement of real-time operation. Conventional architectures for intelligent robotic systems, such as the subsumption architecture[8] and Nasrem architecture[9], do not directly address the problem of reducing uncertainties as well as dealing with unexpected events arid system faults. Furthermore, tile efficacy and efficiency of integrating planning and control in Illulti-resolution time scales are yet to be consolidated.

In this paper, an architecture of intelligent robotic systems, referred to here as GOBS: Goal-Oriented Behavioral Synthesis, is presented for planetary robotic sampling. While connecting sensing and action in real-time, GOBS autonomously synthesizes goal-oriented behaviors or sequences of actions toward the set goals under uncertainties, errors, and faults, through task monitoring and replanning.

GOBS is significant for autonomous robotic tasks in unstructured environments, including planetary robotic sampling, which require robust and fault tolerant behaviors under uncertainties and errors in sensing and actuation as well as in environmental constraints. This paper presents the details of GOBS designed and implemented for typical robotic sampling scenarios on Mars: autonomous soil science where a robot arm trenches soil to examine and deposit soil samples to lander based science instrumentation.

2 GOBS Architecture

GOBS is composed of two major building blocks, the perception and action nets, interconnected in closed loops, as shown in Fig. 1.

The perception net connects logical sensors or features of various levels of abstraction that can be identified by the given sensor system. In Fig. 2, the logical sensors or features that can be extracted from the physical sensors, such as camera, proximity sensor, and tactile sensor, arc organized in a hierarchy, where the logical and physical sensors are depicted respectively as rectangular and elliptical boxes. However, in the perception net, the connections between logical sensors arc further elaborated with their relationships in terms of feature transformation, data fusion, and constraint to be satisfied. For instance, Fig. 3 illustrates the perception net constructed from the logical sensor system of Fig. 2, as follows: The surface-orientation feature may be determined by the distance-to- surface logical sensor based on feature transformation. The same surface-orientation feature may be measured directly by the tactile sensor, such that the feature can be finalized by fusing the two sources of data, one from the clistallceto-surface logical sensor and the other from the tactile sensor. By the same token, the hole-3D-position feature can be determined by fusing the tactile sensor output and the result of feat arc transformation from the hole-21)-J~ositic)Il, surface-c,ricrltatio~l, and clista]lcc-to-surface logical sensors. Furthermore, assuming two holes of the known relative distance, the two hole-3D-position features should be constrained by the known relative distance.

In general, the perception net is formed by the interconnection of logical and physical sells.ors with three types of modules: feature transformation module (FTM), data fusion module (DFM), and

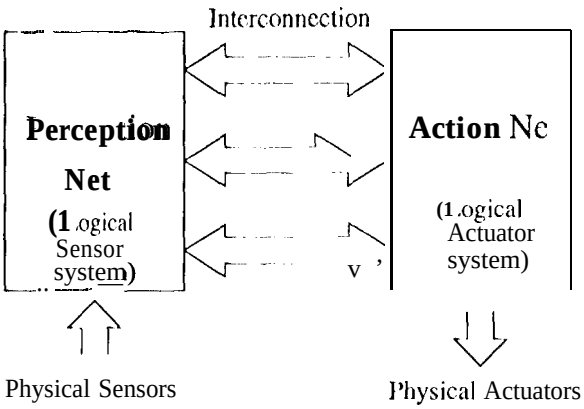

Figure 1: Two Major Building Block, Perception Net and Action Net, of GOBS

constraint satisfaction module (CSM), as shown schematically by Fig. 4. Au FTM transforms a set of primitive features into a more abstract and a higher level of feature. A DFM takes multiple data of a feature to generate an optimal estimate of the feature. A DFM may represent either spatial or temporal data fusion: for the spatial data fusion (s-DFM), data are from the single readings of multiple sensors, while for the temporal data fusion (t-DFM), data are from the multiple readings of a single sensor. Each DFM module is responsible for determining which input data are valid for fusion at the current sensor configuration. A CSM represents system knowledge which imposes a constraint upon a set of feat, ure values.

The output of each logical sensor is a tuple representing the current estimates of corresponding feature value and its uncertainty measure, and is regarded as the current state of the sensor. Then, the net state is defined as the collection of the states of individual logical sensors. The net is operated in such a way that a state change at a logical sensor propagates to adjacent logical sensors, triggering a chain of state changes throughout the net. For example, the state of a logical sensor can be updated by fusing its current state with a new reading from FTM through t-DFM, as schematically depicted in Fig. 4.

Note that the propagation of state change is bidirectional, forward and backward, such that the net automatically updates, and maintains the consistency of, its state not only through the forward propagation of state change but also through the backward propagation of state errors to satisfy constraints. in Fig. 4, the backward signal propagation is explicitly represented by feedback connections from CSMs to the corresponding modules (refer to the detailed lines).

The bidirectional change of net state can be implemented either by distributed computation or by net dynamics, as described in Sec. 3. Through the bidirectional state updating process, the net provides not only the reduction of uncertainties but also the monitoring of errors arid faults, based on which decision-making and replanning take place in the action net. The perception net presents a formal yet, general architecture for sensor fusion and planning. That is, the net can also be used for curbing uncertainties based on active modification of sensing parameters through sensor planning.

The action net consists of a hierarchy of state transition networks of multi-resolution time scales, as shown in Fig. 5. More precisely, the net represents system dynamics in multi-resolution time scales ranging from continuous time to discrete event dynamics, where an action of a higher level of hierarchy is represented by a state transition network of a lower level. The net embeds all the feasible system behaviors in various levels of abstraction. This allows the system to re-plan and control its behaviors efficiently towards the set goals through the feedback of errors, faults, and unexpected events to the various levels of action hierarchy,

The action net cau be interpreted in an analogy to linguistics, The system behaviors that can be generated by the action net are equivalent to tile sentences that can be generated by the given vocabularies and grammar of a language. Applying planning anti control to the action net to generate a goal-oriented behavior for the given task is equivalent to searching for a sequence of grammatical rules to generate a sentence of particular meaning. In this sense, the action net is designed to embed ail the feasible behaviors of the system from which a particular goal-oriented behavior can be searched

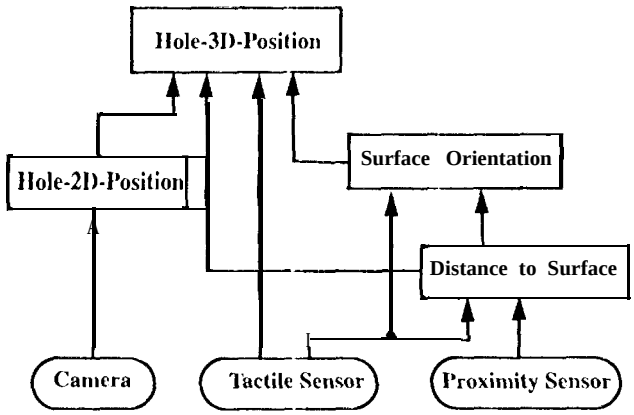

Figure 2: A schematic illustration of a logical sensor system.

for through planning and control.

In summary, GOBS can be considered as a computational knowledge base where concepts are understood by the system through their interconnections and computational dependencies.

3 Uncertainty Management

One of the main features of GOBS is the capability of uncertainty management through its architecture. The reduction of uncertainties may come from data fusion and constraint satisfaction as well as from sensor planning. The processes of data fusion and constraint satisfaction occur when the net changes its state from one equilibrium to another through the forward and backward propagation. The sensor planning occurs when the degree of uncertainty of a state warrants additional data collection with new sensing parameters.

3.1 Uncertainty Representation

The uncertainties of logical sensor outputs are due to the random noise and biases involved in measurement data as well as clue to the biases involved in modeling feature transformations. First, the uncertainties clue to noise are considered. The uncertainties due to biases are handled in the later section concerning the error recovery through calibration.

Although Gaussian randomness of noise as well as independency of data measurements are assumed in sensing, the noise involved in a logical sensor output may not be Gaussian clue to possible nonlinearity in feature transformation. For convenience, however, we assume that noise is bounded by an uncertainty hyper-volume or error ellipsoid, and that the size of error ellipsoid is small enough for a good linear approximation of the nonlinearity around the nominal point in feature transformation. Formally, we represent the uncertainty, dx, of a net variable, x, as an ellipsoid of the following form:

$$dx^t W_x dx \leq 1 \tag{1}$$

where W_x represents a symmetric weight matrix determining the size and shape of the ellipsoid

3.2 Uncertainty Propagation

The uncertainties propagate in the perception net through the input-output relationships of FTM and DFM modules, as well as through the constraints defined by CSM modules. Let us first define the mapping relationship between the input vector, x, and tile output vector, y, of a FTM or a DFM by

$$y = f(x, p) \tag{2}$$

where p represents a parameter vector associated with the module (which may be subject to control for sensor planning, if allowed). Then, the uncertainty propagation through (2) can be approximated

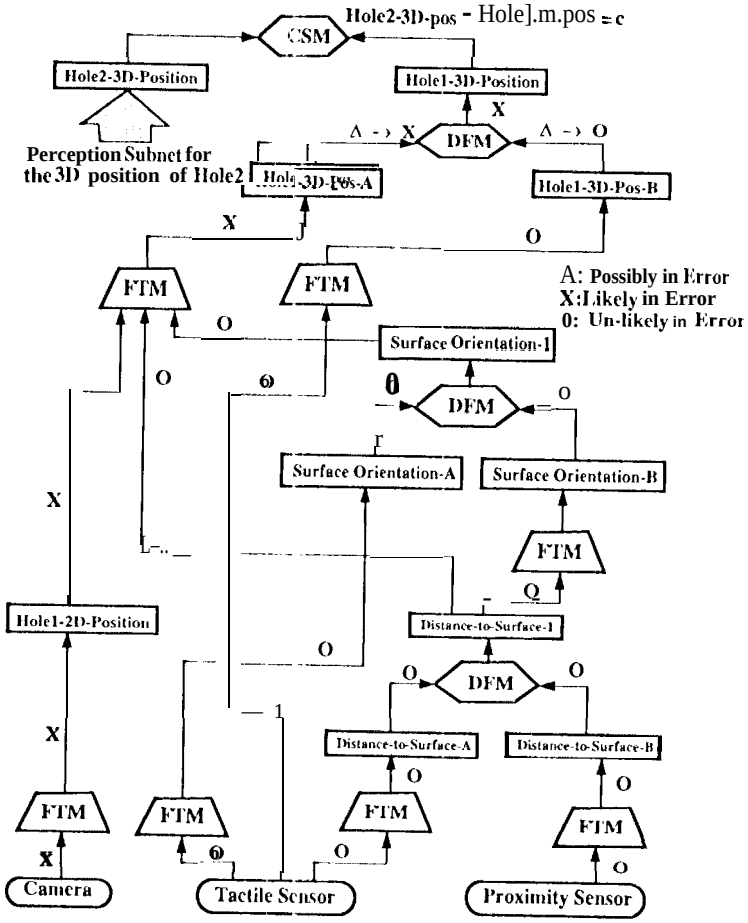

Figure 3: A Perception Net representation of a logical sensor system of Fig. 2. FTM: Feature Transformation Module DFM: Data Fusion Module CSM: Constraint Satisfaction Module

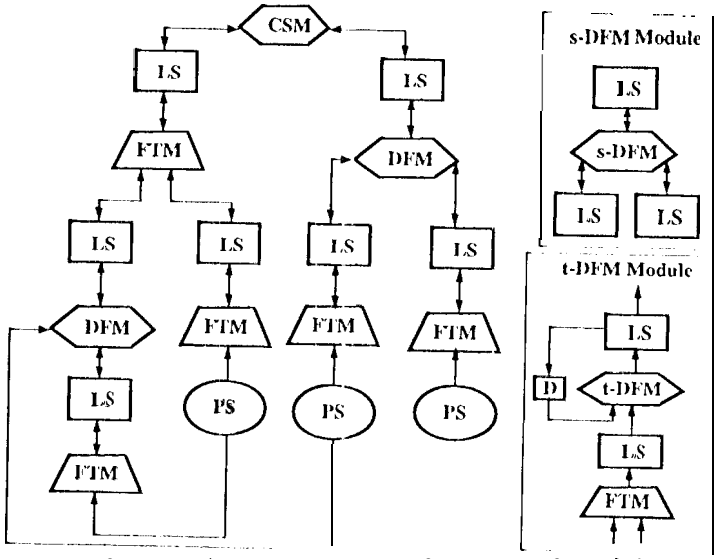

Figure 4: The architecture of perception net composed of 3 types of module: FTM, DFM, and CSM.

as the first order Jacobian relationship with the assumption that f is smooth and dx is small, as follows:

$$y + dy = f(x + dx, p) \approx f(x, p) + \frac{\partial f}{\partial x}\Delta x \tag{3}$$

Therefore,

$$dy \approx \frac{\partial f}{\partial x} dx = J(x, p)dx \tag{4}$$

where $J(x, p)$ represents the Jacobian relationship between dy and dx. The uncertainty of x, represented as an error ellipsoid of (1), can now be propagated to the uncertainty of y, represented as an error ellipsoid in terms of dy, through (4). By substituting $dx = J^{+}(x, p)dy$, obtained from (4), to (l), we have

$$dy^t (J^{+})^t W_x J^{+} dy \leq 1 \tag{5}$$

where J^{+} represents the pseudo-i[lversc of J. Eq. (5) can be rewritten as

$$dy^t W_y dy \leq 1 \tag{G}$$

where the symmetric weight matrix, W_y, is defined as $W_y \equiv (J^{+})^t W_x J^{+}$. Eq. (6) is of the same form as (1). The forward propagation of uncertainties toward the modules of a higher level of hierarchy can be done with the properly defined weight matrices of their input vectors. in the case where the input, x, of a module is composed of two or more vectors, x_1 and x_2, with their respective weight matrices defined as W_{x_1} and W_{x_2}, the weight matrix, W_x, of x can be specified by combining the individual weight matrices W_{x_1} and W_{x_2}, as follows:

$$W_x = \begin{bmatrix} W_{x_1} & 0 \\ 0 & W_{x_2} \end{bmatrix} \tag{7}$$

W_y is a function of x and p, since J is a function of x and p.

3.3 Forward and Backward Propagation . .

The forward propagation of input data, $(x_i, W_{\mathbf{x}_i})$, i= 1,... ,m, through FTM is straightforward once the input-output relationship of FTM is given: the output of FTM, (y, WY), can be obtained directly from (2) and (6). Since DFM can be represented by an input-output relationship), the forward propagation through DFM can also be clone with (2) and (6). The input-output relationship of DFM can be derived from one of the exist ing data fusion methods[6]. However, we present here a new geometric method of data fusion to take into consideration the ellipsoidal bound representation of uncertainties adopted here.

For simplicity, consider the two measurements, $\mathbf{x}_{1m}$ and $\mathbf{x}_{2m}$, defined respectively in the two measurement space, $\mathbf{x}_1$ and $\mathbf{x}_2$, where their uncertainty bounds are defined try the weight matrices, $W_{\mathbf{x}_{1m}}$ and $W_{\mathbf{x}_{2m}}$, respectively. The proposed geometric data fusion method starts with defining the augmented space, z, $z = (\mathbf{x}_1, \mathbf{x}_2)^T$, such that the measurement data, $(\mathbf{x}_{1m}, W_{\mathbf{x}_{1m}})$ and (X2,,,, $W_{\mathbf{x}_{2m}}$) are represented in an augmented space as (z,,,, $W_{\mathbf{z}_m}$), where $\mathbf{z}_m$ = (Xl,,,, $\mathbf{x}_{2m}$)" and $W_{\mathbf{z}_m}$ = Diag[$W_{\mathbf{x}_{1m}}, W_{\mathbf{x}_{2m}}$ J. Then, the problem of fusing $(\mathbf{x}_{1m}, W_{\mathbf{x}_{1m}})$ and $(\mathbf{x}_{2m}, W_{\mathbf{x}_{2m}})$ is equivalent to fine] a point, y, on the constraint manifold, $\mathbf{x}_1 - \mathbf{x}_2 = 0$, defined in the $\mathbf{z}$ space in such a way that the weighted distance between y and $\mathbf{z}_m$, or $\frac{1}{2}\|\mathbf{y} - \mathbf{z}_m\|_{W_{\mathbf{z}_m}}$ is minimum. Once we have an equation for y in terms of $(\mathbf{x}_{1m}, W_{\mathbf{x}_{1m}})$ and (x2,,,, $W_{\mathbf{x}_{2m}}$), WY can be obtained based on the uncertainty propagation method described by (2 - 6).

As a result, the output, y, of DFM with x_i and X_2 as its inputs can be determined as the vector that minimizes $\Sigma\frac{1}{2}\|\mathbf{y} - \mathbf{x}_i\|_{W_{\mathbf{x}_i}}$, as follows:

$$\mathrm{y} = (W_{\mathbf{x}_1} + W_{\mathbf{x}_2})^{-1}(W_{\mathbf{x}_1}\mathbf{x}_1 + W_{\mathbf{x}_1}\mathbf{x}_2) \tag{8}$$

The uncertainty bound, WY, associated with y can be obtained by applying (4)(5)(6) to (8):

$$W_{\mathbf{y}} = [AA^T + BB^T]^{-2}[AW_{\mathbf{x}_1}A^T + BW_{\mathbf{x}_2}B^T] \tag{9}$$

$$\text{where } A \triangleq [W_{x_1} + W_{\mathbf{x}_2}]^{-1} W'_{x_1}$$

$$B \triangleq [W_{\mathbf{x}_1} + W_{\mathbf{x}_2}]^{-1} W_{\mathbf{x}_2}$$

Note that (8) is of the same form to the maximum Bayesian posteriori probability with the Gaussian assumption of noise.

The backward propagation process starts with a CSM. For instance) consider that the two logical sensor output vectors, x and y, are constrained by $Ax + B\mathbf{y} = c$. CSM evaluates whether the current estimates, $\mathbf{x}_f$ and $\mathbf{y}_f$, of x and y from the forward process satisfy the given constraint. If not, CSM updates $(\mathbf{x}_f, W_{\mathbf{x}_f})$ and $(\mathbf{y}_f, W_{\mathbf{y}_f})$, where $W_{\mathbf{x}_f}$ and $W_{\mathbf{y}_f}$ are the weight matrices associated with $\mathbf{x}_f$ and $\mathbf{y}_f$, respectively, into $(\mathbf{x}_b, W_{\mathbf{x}_b})$ and $(\mathbf{y}_b, W_{\mathbf{y}_b})$ in such a way that $\mathbf{x}_b$ and $\mathbf{y}_b$ satisfy the constraint, The derivation of a mathematical formula for this process is same as that of DFM: the concepts of CSM and DFM can be unified under the proposed geometric method of data fusion.

To show this, let us first define the augmented space $\mathbf{z}$ with x and y, $\mathbf{z} = (x, y)^T$. Then, the constraint of CSM can be represented as a manifold in the augmented space. Furthermore, $(x_f, W_{\mathbf{x}_f})$ and $(\mathbf{y}_f, W_{\mathbf{y}_f})$ can be represented in the augmented space as $(\mathbf{z}_f, W_{\mathbf{z}_f})$, with $\mathbf{z}_f = (\mathbf{x}_f, \mathbf{y}_f)^T$ and $W_{\mathbf{z}_f} = \mathrm{Diag}[W_{\mathbf{x}_f}, W_{\mathbf{y}_f}]$. Finally, by selecting a vector, $\mathbf{z}_b, \mathbf{z}_b = (\mathbf{x}_b, \mathbf{y}_b)^T$, on the constraint manifold in such a way that the weighted distance from $\mathbf{z}_b$ to $\mathbf{z}_f$, $\frac{1}{2}\|\mathbf{z}_b - \mathbf{z}_f\|_{W_{\mathbf{z}_f}}$, is minimum, we can obtain $(\mathbf{x}_b, W_{\mathbf{x}_b})$ and $(\mathbf{y}_b, W_{\mathbf{y}_b})$, as follows:.

Miu $\frac{1}{2}\|\mathbf{z}_b - \mathbf{z}_f\|_{W_{\mathbf{z}_f}}$ with $\mathbf{z}_b$ on the constraint manifold implies Min $\frac{1}{2}(\|\mathbf{x}_b - \mathbf{x}_f\|_{W_{\mathbf{x}_f}} + \|\mathbf{y}_b - \mathbf{y}_f\|_{W_{\mathbf{y}_f}}$ with $A\mathbf{x}_b + B\mathbf{y}_b = c$. Since $\mathbf{y}_b = B^{-1}(c - A\mathbf{x}_b)$ or $\mathbf{x}_b = A^{-1}(c - B\mathbf{y}_b)$ on the constraint manifold, Min $\frac{1}{2}(\|\mathbf{x}_b - \mathbf{x}_f\|_{W_{\mathbf{x}_f}} + \|\mathbf{y}_b - \mathbf{y}_f\|_{W_{\mathbf{y}_f}})$ with $A\mathbf{x}_b + B\mathbf{y}_b = c$ can be expressed in the $\mathbf{x}_b$ space as Miu $\frac{1}{2}(\|\mathbf{x}_b - \mathbf{x}_f\|_{W_{\mathbf{x}_f}} + \|B^{-1}(c - A\mathbf{x}_b) - \mathbf{y}_f\|_{W_{\mathbf{y}_f}})$, or in the $\mathbf{y}_b$ space as Min $\frac{1}{2}(\|A^{-1}(c - B\mathbf{y}_b) - \mathbf{x}_f\|_{W_{\mathbf{x}_f}} + \|\mathbf{y}_b - \mathbf{y}_f\|_{W_{\mathbf{y}_f}})$. Then, $\mathbf{x}_b$ and $\mathbf{y}_b$ can be obtained by solving the corresponding minimization problems in $\mathbf{x}_b$ and $\mathbf{y}_b$ space, as follows:

$$\mathbf{x}_b = [W_{\mathbf{x}_f} + (B^{-1}A)^T W_{\mathbf{y}_f}(B^{-1}A)]^{-1}[W_{\mathbf{x}_f}\mathbf{x}_f + (B^{-1}A)^T W_{\mathbf{y}_f}(B^{-1}c - \mathbf{y}_f)] \quad (10)$$

$$\mathbf{y}_b = [W_{\mathbf{y}_f} + (A^{-1}B)^T W_{\mathbf{x}_f}(A^{-1}B)]^{-1}[W_{\mathbf{y}_f}\mathbf{y}_f + (A^{-1}B)^T W_{\mathbf{x}_f}(A^{-1}c - \mathbf{x}_f)] \quad (11)$$

Eqs. (10) and (11) define the input-output mappings of the form, $\mathbf{x}_b = A_{\mathbf{x}}\mathbf{x}_f + B_{\mathbf{x}}\mathbf{y}_f + K_{\mathbf{x}}$ and $\mathbf{y}_b = A_{\mathbf{y}}\mathbf{x}_f + B_{\mathbf{y}}\mathbf{y}_f + K_{\mathbf{y}}$, where $K_{\mathbf{x}}$ and $K_{\mathbf{y}}$ are constant. Therefore, by applying Eqs. (7), (8) and (9) of uncertainty propagation to (10) and (11), we have the same result as (9) with $A = AX$ and $B = B_{\mathbf{x}}$ or $A = A_{\mathbf{y}}$ and $B = B_{\mathbf{y}}$

Once $\mathbf{x}_f$ and $\mathbf{y}_f$ are updated to $\mathbf{x}_b$ and $\mathbf{y}_b$, the updated vectors, $\mathbf{x}_b$ and $\mathbf{y}_b$, in turn, impose constraint on the lower level processes in the hierarchy. 'Thus, the same backward propagation defined for CSM can be applied to the subsequent process, except that $\mathbf{x}_b$ and $\mathbf{y}_b$ are associated with uncertainty matrices. This case is equivalent to the CSM problem but with the uncertainty of c represented by $d_c^T W_c d_c = 1$. In this case, $\mathbf{x}_b$ and $\mathbf{y}_b$ can be obtained by the same way as (10) and (11) as if c has no uncertainty. However, in order to compute $W_{\mathbf{x}_b}$ and $W_{\mathbf{y}_b}$, $\mathbf{x}_b$ and $\mathbf{y}_b$ should be represented as a function of $\mathbf{x}_f, \mathbf{y}_f$ arid c so as to consider the effect of W_c on $W_{\mathbf{x}_b}$ and $W_{\mathbf{y}_b}$: $\mathbf{x}_b = A_{\mathbf{x}}\mathbf{x}_f + B_{\mathbf{x}}\mathbf{y}_f + C_{\mathbf{x}}c$ and $\mathbf{y}_b = A_{\mathbf{y}}\mathbf{x}_f + B_{\mathbf{y}}\mathbf{y}_f + C_{\mathbf{y}}c$, such that $W_{\mathbf{x}_b}$ and $W_{\mathbf{y}_b}$ can be computed based on Eqs. (7),(8), and (9), with $[AX, B_{\mathbf{x}}, C_{\mathbf{x}}]$ and $[A_{\mathbf{y}}, B_{\mathbf{y}}, C_{\mathbf{y}}]$ defined as Jacobians.

4 Error Monitoring and Recovery

1 DFM ant{ CSM of the perception net make it possible to monitor errors, i.e., biases and faults in sensing and action. Upon the identification of biases and faults in sensing and action, the action net invokes error recovery and repairment actions. In GOBS, error monitoring and recovery consist of 1) error detection, 2) error identification, and 3) sensor calibration and action replanning for error recovery and repairment.

4.1 Error Detection

Errors may be detected from the following information:

1. The discrepancy between the planned and measured states,
2. The inconsistency among the input data of DFM.

3. The violation of constraints at CSM.

To clarify the meaning of discrepancy, inconsistency, and violation, we need to quantitatively define the thresholds that separate the effect of biases ant] faults from that of uncertainties.

The inconsistency among the input data of DFM can be evaluated based on the ellipsoidal representation of uncertainty bounds: The input data of DFM are said to be inconsistent if the ellipsoidal uncertainty bounds of input data have no common intersection. More formally, the existence of a common intersection among the input data, (x_1, W_1), (x_2, W_2), ... (x_n, W_n), where W_i is the weight matrices associated with $\mathbf{x}_i$ for i= 1 ,... n, can be evaluated by the following rule:

> If $\|\mathbf{x} - \mathbf{x}_i\|_{W_i} \leq 1$, for i = 1 ,....,n with $X = (\Sigma W_i)^{-1} \Sigma W_i \mathbf{x}_i$, Then, there exists a common intersection among the ellipsoidal uncertainty bounds of $\mathbf{x}_i$, i= 1,...,n. Otherwise, there exists no common intersection.

Similarly, the violation of constraint at CSM by $\mathbf{z}_f = (\mathbf{x}_f, \mathbf{y}_f)^T$ can be detected by checking whether the vector, $\mathbf{z}_b, \mathbf{z}_b = (\mathbf{x}_b, \mathbf{y}_b)^T$, on the constraint manifold that minimizes the weighted distance from $\mathbf{z}_b$ to $\mathbf{z}_f$, $\|\mathbf{z}_b - \mathbf{z}_f\|_{W_{\mathbf{z}_f}}$, is inside the uncertainty ellipsoid of $\mathbf{z}_f$:

> If $\|\mathbf{z}_b - \mathbf{z}_f\|_{W_{\mathbf{z}_f}} \leq 1$, then, $(\mathbf{x}_f, \mathbf{y}_f)$ does not violate the constraint, Otherwise, $(\mathbf{x}_f, \mathbf{y}_f)$ violates the constraint.

The discrepancy between the planned and measured states can also be detected by checking whether there is a common intersection between the uncertainty ellipsoids of planned and measure states, where the uncertainty ellipsoid of planned state can be determined by tile expected uncertainty involved in plan execution.

4.2 Error Identification

Upon the detection of errors, there needs to identify the source of errors. When more than two input data arc involved in DFM, we can check which input data is isolated from the rest in terms of sharing a common intersection. in general, for a DFM with multiple input data, it is possible to identify groups of input data that share a common intersection (based on the method presented above). The input data that belongs to the group of single or small number of members may be considered as a likely source of error.

When error is detected in the input data, $(\mathbf{x}_f, W_{\mathbf{x}_f})$ and $(\mathbf{y}_f, W_{\mathbf{y}_f})$, of CSM, wc can check whet her $\mathbf{x}_b$ and $\mathbf{y}_b$ are inside the uncertainty ellipsoids of $(\mathbf{x}_f, W_{\mathbf{x}_f})$ and $(\mathbf{y}_f, WY,)$, respectively. That is, if $\|\mathbf{x}_b - \mathbf{x}_f\|_{W_{\mathbf{x}_f}} > 1$ or if $\|\mathbf{y}_b - \mathbf{y}_f\|_{W_{\mathbf{y}_f}} > 1$, then $\mathbf{x}_f$ or $\mathbf{y}_f$ may be a likely source of error.

Further isolation of error sources can be done through the net hierarchy. By applying the above error detection method to DFMs and CSMs distributed in the net, those logical sensors associated with DFMs and CSMs can be classified either likely-in-error, unlikely-in-error, or possibly-irl-error. Then, these classifications arc propagated through the net to extend the classifications to other logical sensors connected through the hierarchy. The cross-checking of these classifications propagated through the net hierarchy provides further isolation of errors, as shown in Fig 3. We can extend the propagation and cross-checking of classifications to the action net, since the discrepancy between the planned and measured states provides additional error detection. If the above DFM and CSM based error identification method fail to isolate error sources, sensor planning or error-isolation action should take place in the action net for complete isolation of errors,

4 . 3 Error Recovery

Once error sources arc isolated, then the system must take actions to repair the errors and to recover from the errors. Two types of actions can take place: 1) Calibration of sensors to eliminate biases. 2) Replanning the actions to reach the desired goal state under errors. For the first, a predefined sensor calibration routine for the sensor in error will be invoked by the action net. For the second, the action net replans the task based on the GOBS modified according to the isolated errors,

5 Planetary Robotic Science Sampling: Soil Science

Autonomous soil science includes the trenching of planetary soil by a robot to collect and analyze subsurface soil samples, Autonomous soil science is composed of the following activities: trenching site designation by scientists, visual and tactile verification of trenching site by a robot, planning of optimal trenching trajectories with measured s property, adaptive trenching, planning of optimal scooping trajectory, and adaptive scooping. Undoubtedly, uncertainty management as well as error monitoring and recovery play an important role,

More details arc described by the following steps:

Step 1:
Scientists choose a desirable trenching location, length and depth at the ground station in interaction with the monitoring system based on the down-linked stereo camera images and a mouse-based user interface.

step 2:
With the starting point, trenching length and depth data from the ground station, GOBS explores the trenching surface by means of touch and tactile exploration with the robot hand in order to accurately localize trenching surface and measure surface rigidity or resistance. These data are used to construct the desired trenching trajectories with optimal arm Configurations.

Step 3:
The arm moves to the initial contact point and follows the preset trenching trajectories with the optimal arm configurations. Trenching is started by cutting the soil at a prescribed depth with a predefined end-effector orientation and swath towards the lander. Excavated soil is deposited at the back end of the trench. GOBS responds to soil and trenching variables to meet the science goals: GOBS monitors the relationship between the measured force and motion during trenching to detect abnormalies and change the trenching trajectories and arm configurations in adaption to variations and abnormalies. GOBS also makes sure of obtaining the required scientific measurements with soil temperatures, cutting forces, cutting speed, etc. Trenching continues nominally until a prescribed depth is reached.

step 4:
At the prespecified depth, or in the case where soil sampling is dictated by GOBS due to hard soil or rock (indicated by sensors), the arm collects soil sample by scooping operations and deposits the sample in the Lander instrument bin. GOBS verifies that the arm actually has the soil sample in its end-effector. GOBS invokes corrective actions, if necessary.

The GOBS architecture to implement the above steps is illustrated in Figs. 6, 7

5.1 GOBS Operations

Through the perception net, shown in Fig. 3, GOBS explicitly manages uncertainties: Uncertainties associated with individual logical sensors (depicted as circle) arc propagated through such functional modules as DFM, FTM (trapezoidal shape), and CSM (double hexagon). The values associated with logical sensors are updated through forward and backward process of reaching an equilibrium point.

In the perception net, the reduction of uncertainty in locating the scoop at the designated trenching site is highlighted by the data fusion with the joint encoders, the stereopsis with a marker, and the tactile exploration, as well as the constraint from the trenching plane. in the action net, the top level of the action net of GOBS for soil trenching and scooping is shown in Fig. 7, where actions arc depicted by boxes while states are depicted by (double) circles. The lower level of the action net includes the details of actions defined at the higher level: e.g., the adaptive trenching state transition network in Fig. 8.

Example: Error Monitoring and Recovery-1

The arm is commanded to move to the preplanned approach point. There can occur the following two scenarios:

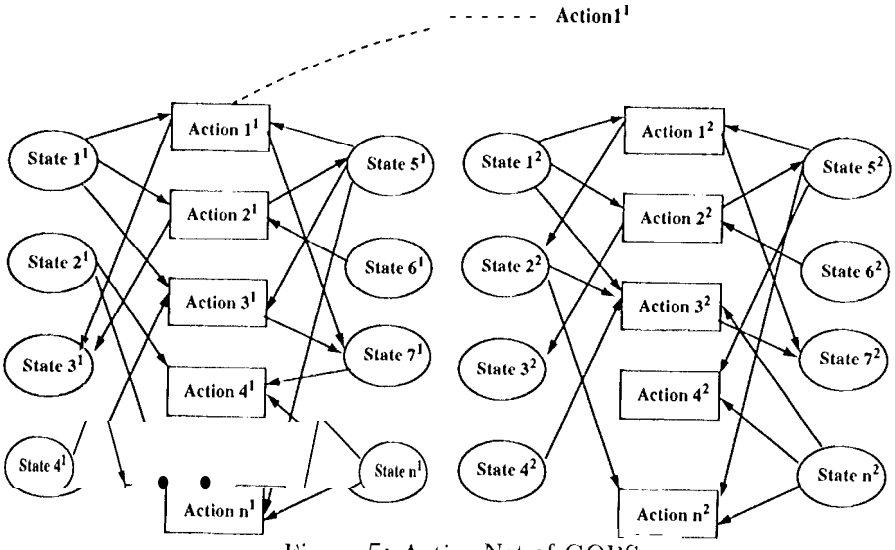

Figure 5: Action Net of GOBS

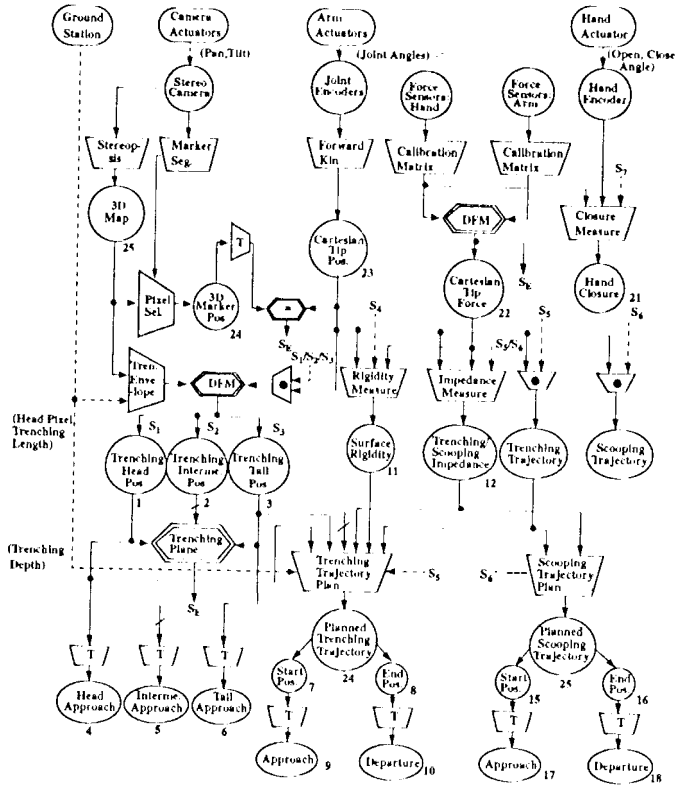

Figure 6: Perception Net of GOBS for Soil Science

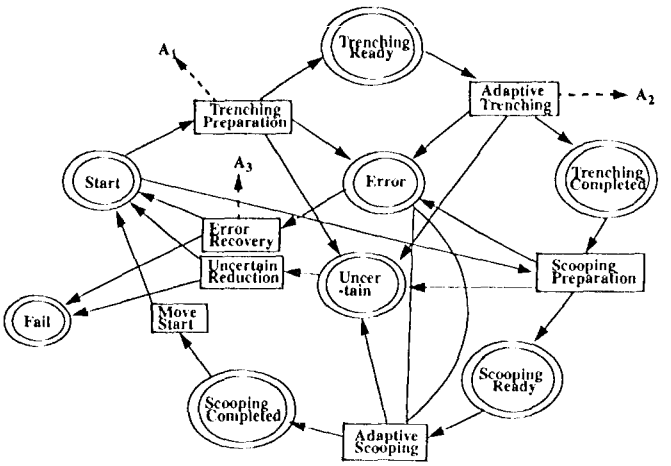

Figure 7: Action Net of GOBS for Soil Science

1. the task is successfully completed, i.e., the sensor reading coincides with the planned position (within some allowed error range)

2. the task is not successfully completed, within the expected time), i.e., the system fails to reach the planned position.

Now, we can check the actual end-effector position by using 3-D marker.

In the first case, if the 3D-marker reading coincides with the encoder reading, then the system is in a normal operational mode, and data fusion can occur. However, if the 3D-marker reading does not coincide with the encoder reading, then, we can say that either the encoder is biased or the marker reading (vision) is biased. The encoder bias can be calibrated through the arm calibration procedure (using a zero position or a reference position and potentiometer). Once the encoder bias is calibrated, then we know whether the inconsistency is due to the bias in marker reading. If so, we need to follow the vision calibration procedure.

In the second case, if the 3D-marker reading coincides with the encoder reading, then it is likely that the actuator or controller is in fault. In this case, the system can make a sense of that motion (for individual joints) to further isolate which actuator is in error. If the 3D-marker reading does not coincide with the encoder reading, the error may be either in encoder and/or in actuator and/or in marker reading. Encoder calibration and the actuator fault isolation routines are necessary for further identification of the problem.

Example: Error Monitoring and Recovery-2

During trenching, the abnormal encoder reading or impedance measure indicate that the trenching is not progressing well. This may be the case where the scoop is stuck at the rigid soil or underground rock site or the failure of adaptive impedance control for trenching. The failure of adaptive impedance control may come from actuators, encoders, power supply, controllers, or force sensor. The abnormal impedance readings means the violation of the preset force and position error relationship. In the case where the impedance measure is normal but encoder readings indicates the jamming situation, then we apply the discrete event control to modify the trenching trajectory. In the case where the impedance measure is not normal, then we perform a series of actions to identify the source exactly. For instance, the routines for the calibration of encoders, force sensors and for checking actuator performance, etc. Based on the above, the problem can be pin-pointed.

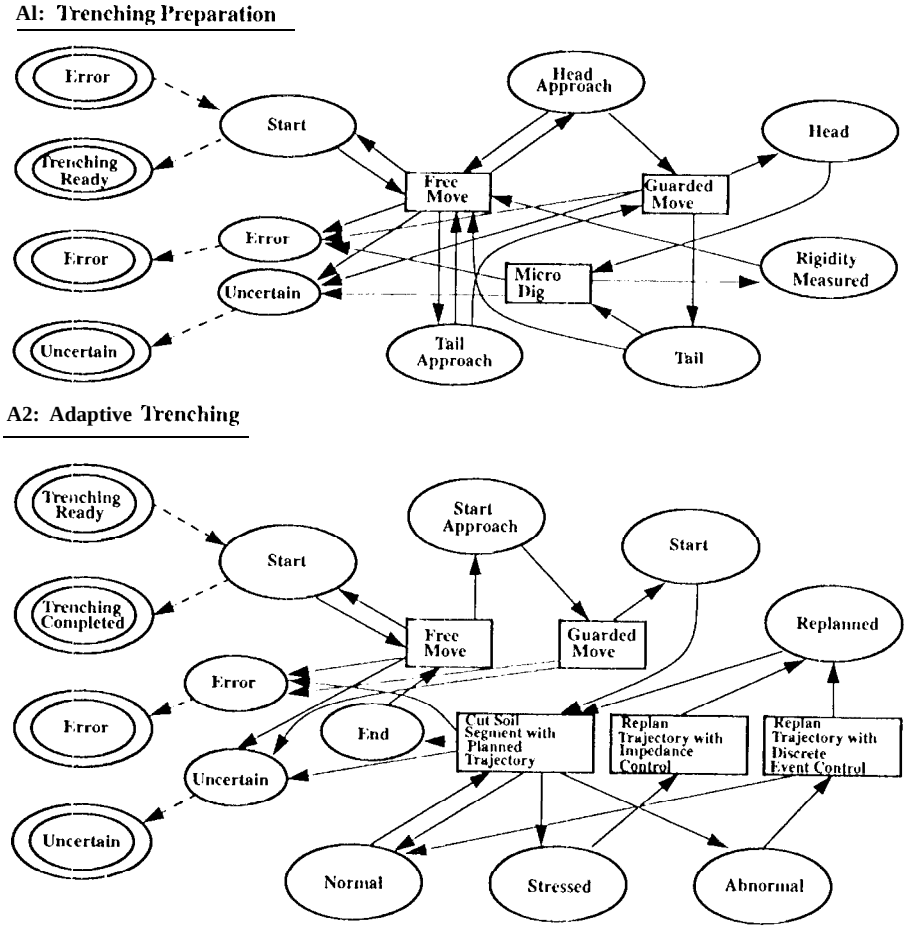

Figure 8: A_1, A_2, and A_3 of Fig. 7

6 Conclusion

The proposed GOBS architecture provides a formal mechanism of integrating sensing, knowledge, and action in real-time for intelligent robots. The architecture emphasizes uncertainty management as well as error monitoring and recovery such that the system can provide robots with the capability of generating goal-oriented, yet robust and fault tolerant, behaviors. The proposed geometric method for uncertainty management and error monitoring through the perception net is novel and powerful due to its systematic method. One might find it interesting to compare it with the existing probability network. 'The perception net provides a more general but formal way of accomplishing sensor fusion and planning. 'The proposed GOBS architecture also serves as a general intelligent control architecture that can be applicable to the control of complex systems including spacecraft and power plants,

Future work includes the continuous implementation of the GOBS and the evaluation of GOBS in comparison with existing architectures, especially, in terms of the power or power-to-weight ratio of intelligence measure introduced in the beginning of this paper.

References

[1] Ren C. Luo, Min-Hsiung Lin, and Ralph S. Scherp, "Dynamic Multi-Sensor Data Fusion System for intelligent Robots," IEEE Journal of Robotics and Automation, Vol. 4, No. 4, pp. 386-385, August 1988.

[2] Hans F'. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," AI Magazine, pp. 61-74, Summer 1988.

[3] Scott W. Shaw, Rui J. P. deFigueiredo, and Kumar Krishen, " Fusion of Radar and Optical Sensors For Space Robot Vision," The proceeding of IEEE conference on Robotics and Automation, pp. 1842-1846, 1988.

[4] Tom Henderson and Esther Shilcrat, "Logical Sensor System," Journal of Robotic System, 1 (2), pp. 169-193, 1984.

[5] '1 '.1.. Huntsberger and S. N. Jayaramamurthy, " A Frame Work for Multi-Sensor Fusion in the Presence of Uncertainty," Proceedings of Spatial Reasoning and Multi-Sensor Fusion Workshop, pp. 345-350, 1987.

[6] Hugh F. Durrant-Whyte, "Sensor Models and Multi-Sensor integration," IEEE Journal of Robotics and Automation, 1987.

[7] H. F. Duran-White, "A Bayesian approach to optimal sensor placement ," International Journal of Robotics Research, 9(5):70-88, 1990.

[8] Rodney Brooks, "A Robust layered control system for a mobile robot", IEEE Journal of Robotics and Automation, Vol. RA-2, NO. 1, March 1986

[9] James S. Albus, " Brains, behavior, and robotics", BYTE Books, Peterborough, NH, 1981

Acknowledgement

The research described in this paper was in part carried out by the Jet Propulsion laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.